Chapter 3

Methodology

The block diagram of our project is shown below:

Its consists of two parts

1. Software
2. Hardware

3.1. Block Diagram

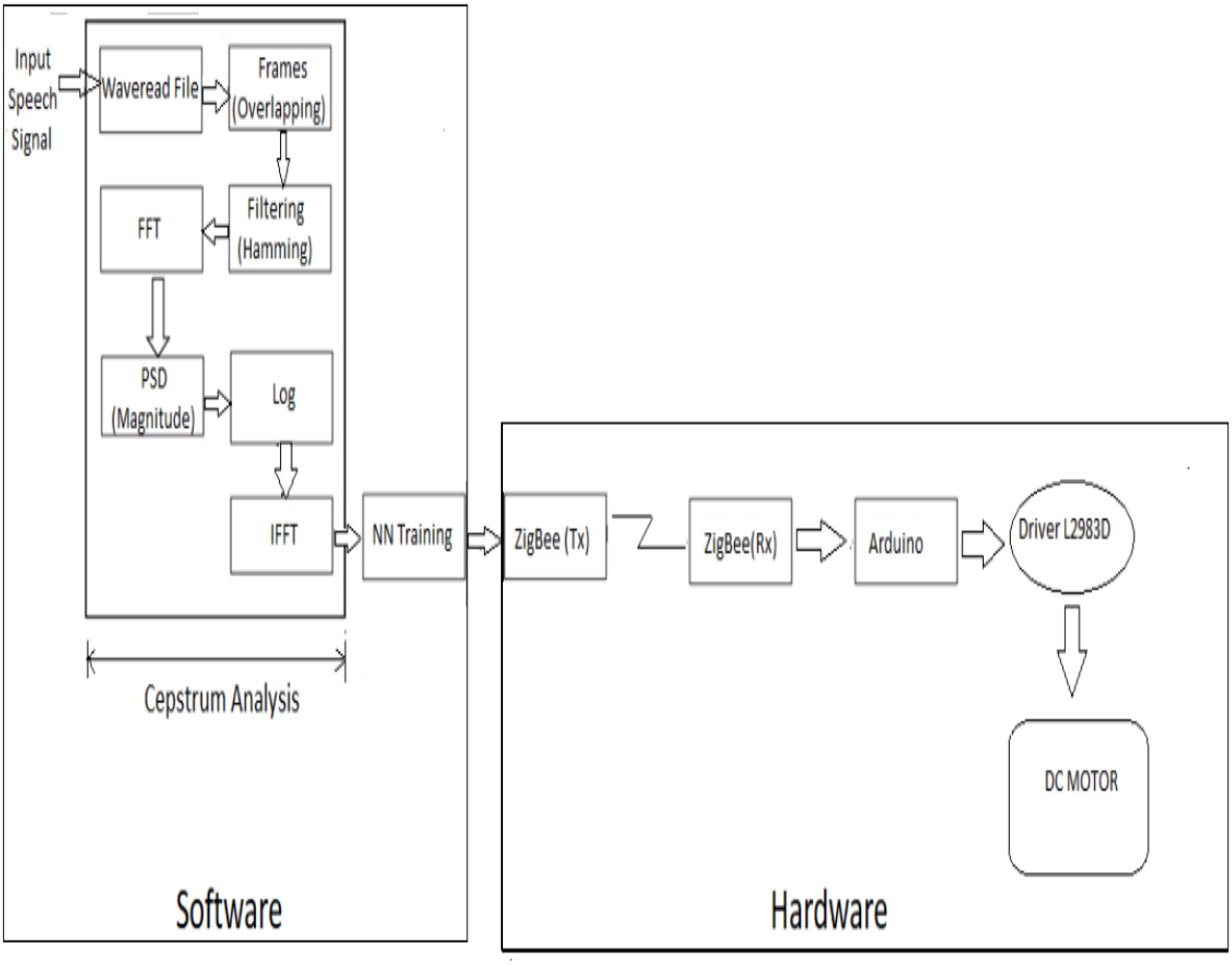

Fig. 3.1 Block Diagram

3.2. Description

3.2.1. Capturing the signal

In the processing part the first step is to capture the signal. The signal from a human is an analog signal. When storing the speech to a computer, the analog signal needs to be digitized. So the first component of speech processing is speech signal measurement. When people talk to a microphone, the analog signal pressurizes the air, then the analog electric signals goes to the microphone. Analog speech digitization has two steps: Sampling and Quantification.

Sampling-The sampling of analog signal is based on sampling theorem. The sampling theorem states that if f_m is the maximum frequency component in the analog signal, then the information present in the signal can be represented by its sampled version provided the number samples

taken per second is greater than or equal to twice the maximum frequency component. The number of samples/second is more commonly termed as sampling frequency f_s. According to sampling theorem, f_s should be greater than or equal to 2 f_m.

Quantification-Quantification means that the signal that is discrete in time domain but continuous in amplitude is turned into a signal that is discrete in amplitude. When doing this the amplitude values are cut into limited extent. According to the sampling precision, the samples that are in one extent will be given the same amplitude value. The dynamic range of the voice is limited by quantification, its unit is bit.

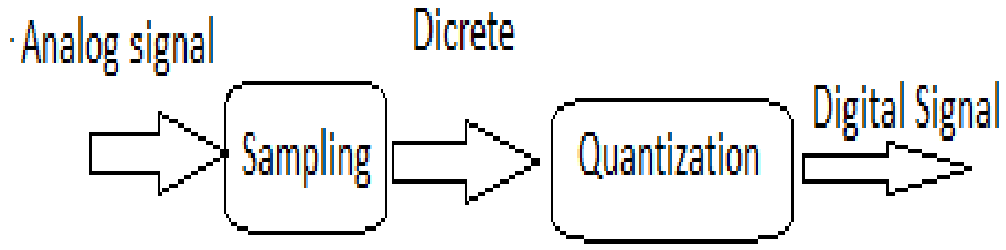

Fig. 3.2.1 Pre-processing

3.2.2. Pre-processing

3.2.2.1. Silence Removal

A Silence Removal for Speech Processing is used to remove the DC offset value from the signal after silence removal process. Silence removal remain part of many applications such as speaker and speech recognition. The speech signals usually contain many areas of silence or noise. The silence signal is useless for recognition, because it contains no information. And if we keep the silence signal it will make the processing signal larger and take more time and space when getting information or features from the signal. So only the signal part that contains the actual speech segments is useful for recognition.

3.2.2.2. Normalization

Normalization is a method for adjusting the volume of audio files to a standard level, as different recording levels can cause the volume to vary greatly from word to word. The recording sound samples with different volumes and possibly some DC offset, should naturally not influence the detection system. A simple way of avoiding this is to normalize the signaling some way, e.g. scaling, and offsetting the signal so that it falls between levels -1 and 1. So a normalization is

needed and can be applied before any other processing. The features that we will extract later on, e.g. the Mel-Frequency transform, depend on the power of the signal. This implies that speaking loudly will be seen differently than quietly. By normalizing there coding signal, this effect can be reduced.

3.2.2.3. Pre-emphasis

Pre-emphasis is a very simple signal processing method which is used to increase the amplitude of high frequency bands and decrease the amplitudes of lower bands, this process boosts the energy in high frequency which will give more information for the further process and also will improve the signal for same frequency. Then this signal is framed .The width of the signal is generally about 30 ms with an overlap about 20 ms that's mean it's shifted generally 10 ms. Each frame contains N number of sample points of the speech signal. If the frame is short, then it will not get the enough samples to estimate the reliable spectral that is the reason why we need to select the frame signal into 20-40mS frame which will gives throughout

3.2.2.4. Windowing

An audio signal is an unstable signal, meaning that the statistical properties across time are not constant. However in a very short period of time the properties can be regarded as constant .Then a short piece of signal is cut out of the whole speech signal. This is done by multiplying the speech samples with a windowing function to cut out a short segment of the speech signal .The time for which the signal is considered for processing is called a window, and the data acquired in a window is called a frame. Features are extracted once every Mms, which is called frame rate, while the window duration is N ms. Thus two consecutive frames have overlapping areas. The overlapping segments are used for speech analysis. The choosing of frame length and frame shift are very important, as it can have different effect on eliminating noise as well.

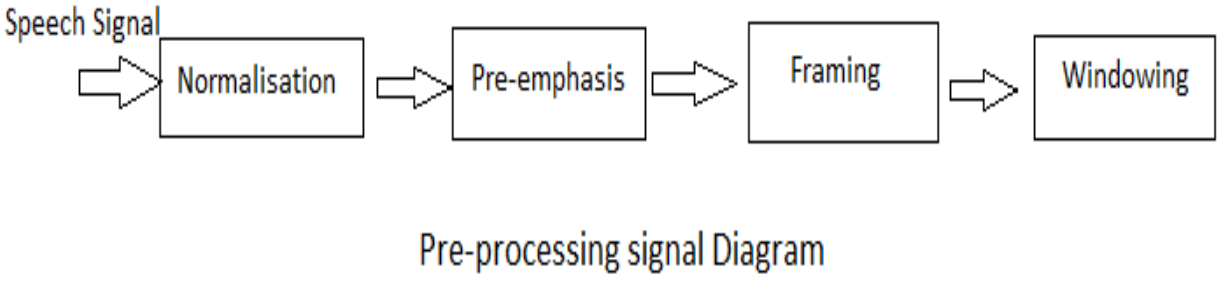

Fig.3.2.2.4 Pre-processing signal diagram

3.2.3. Feature Extraction

After the pre-processing we could have fed each 'raw' audio sample to a neural network, to let the neural network learn to recognize a speaker or speech based on that. This could work for speaker recognition at least, but this was not feasible for a number of reasons:

- Once a signal has been sampled, we have huge amounts of data every second.
- Feeding each sample to a neural network, that is one input to the neural network for each sample, which means a huge number of inputs to the NN.
- We can only capture the properties of the signal that are important for recognition instead of using all the samples.
- The spectrogram contains much of the information we need.
- By extracting the features, the same features can be used by multiple tasks.

The features are much lower in number, than the raw audio samples. They are an extremely more refined and compressed input to the neural network. So based on theoretical and practical operation, we will extract features from the frames. The Merriam-Webster dictionary explains the word feature e.g. as: “a prominent part or characteristic”, or by an example: 'Her eyes are her best feature”. That is, we wish to extract some prominent characteristics in the frames of an audio sample of a speaker, or words (speech).When doing speaker recognition the features need to react the information of every speaker as much as possible. In another way we can say that feature extraction is to transform the input waveform into a sequence of acoustic feature vectors, each vector representing the information in a small time window of the signal. Feature extraction transforms high dimensional input signals into lower dimensional vectors. That is, the large number of audio samples, in our case 11025 each second, is translated into a small number of features that are somehow characteristic of some speaker/speech

3.2.3.1. LPC (Linear Predictive Code) and MFCC

LPC analyzes the speech signal by estimating the formants, removing their effects from the speech signal, and estimating the intensity and frequency of the remaining buzz. The process of removing the formants is called inverse filtering, and the remaining signal after the subtraction of the filtered modeled signal is called the residue. LPC is frequently used for transmitting spectral envelope information, and as such it has to be tolerant of transmission errors. Transmission of the filter coefficients directly (see linear prediction for definition of coefficients) is undesirable, since

they are very sensitive to errors. In other words, a very small error can distort the whole spectrum, or worse, a small error might make the prediction filter unstable.

Importance of linear prediction analysis in speech

Speech signal is produced by the convolution of excitation source and time varying vocal tract system components. These excitation and vocal tract components are to be separated from the available speech signal to study these components independently. For deconvolving the given speech into excitation and vocal tract system components, methods based on homomorphic analysis like cepstral analysis are developed. As the cepstral analysis does the deconvolution of speech into source and system components by traversing through frequency domain, the deconvolution task becomes computational intensive process. To reduce such type of computational complexity and finding the source and system components from time domain itself, the Linear Prediction analysis is developed.

MFCC:

The extraction of the best parametric representation of acoustic signals is an important task to produce a better recognition performance. The efficiency of this phase is important for the next phase since it affects its behavior. MFCC is based on human hearing perceptions which cannot perceive frequencies over 1Khz. In other words, in MFCC is based on known variation of the human ear's critical bandwidth with frequency. MFCC has two types of filter which are spaced linearly at low frequency below 1000 Hz and logarithmic spacing above 1000Hz. A Subjective pitch is present on Mel Frequency Scale to capture important characteristic of phonetic in speech. MFCC consists of seven computational steps. Each step has its function and mathematical approaches as discussed briefly in the following:

Step 1: Pre–emphasis

This step processes the passing of signal through a filter which emphasizes higher frequencies. This process will increase the energy of signal at higher frequency.

Step 2: Framing

The process of segmenting the speech samples obtained from analog to digital conversion (ADC) into a small frame with the length within the range of 20 to 40 msec. The voice signal is divided into frames of N samples.

Step 3: Hamming windowing

Hamming window is used as window shape by considering the next block in feature extraction processing chain and integrates all the closest frequency lines. The Hamming window equation is given as:

If the window is defined as

$W(n), 0 = n = N-1$ where

N = number of samples in each frame

Y[n] = Output signal

X (n) = input signal

W (n) = Hamming window, then the result of windowing signal is shown below:

$$Y(n)=X(n)*W(n)$$

Step 4: Fast Fourier Transform

To convert each frame of N samples from time domain into frequency domain. The Fourier Transform is to convert the convolution of the glottal pulse U[n] and the vocal tract impulse response H[n] in the time domain. This statement supports the equation below:

$$Y(w)=FFT[h(t)*X(t)]=H(w)*X(w)$$

Step 5: Mel Filter Bank Processing

The frequencies range in FFT spectrum is very wide and voice signal does not follow the linear scale. Each filter's magnitude frequency response is triangular in shape and equal to unity at the centre frequency and decrease linearly to zero at centre frequency of two adjacent filters. Then, each filter output is the sum of its filtered spectral components.

Step 6: Discrete Cosine Transform

This is the process to convert the log Mel spectrum into time domain using Discrete Cosine Transform (DCT). The result of the conversion is called Mel Frequency Cepstrum Coefficient. The set of coefficient is called acoustic vectors. Therefore, each input utterance is transformed into a sequence of acoustic vector.

Step 7: Delta Energy and Delta Spectrum

The voice signal and the frames changes, such as the slope of a formant at its transitions. Therefore, there is a need to add features related to the change in cepstral features over time. 13 delta or velocity features (12 cepstral features plus energy), and 39 features a double delta or acceleration feature are added.

3.2.4. Neural Networks

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way of biological nervous system, such as the brain process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing element working in unison to solve specific problems. An ANN is configured for a specific application such as pattern recognition or data classification through a learning process. Neural networks are nonlinear data driven self-adaptive powerful tools for modeling, especially when the underlying data relationship is unknown. In artificial neural networks have been applied successfully to speech recognition, image analysis and adaptive control, in order to construct software agents (in computer and video games) or autonomous robots. Most of the currently employed artificial neural networks for artificial intelligence are based on statistical estimations, classification optimization and control theory. The cognitive modeling field involves the physical or mathematical modeling of the behaviour of neural systems, ranging from the individual neural level (e.g. modeling the spike response curves of neurons to a stimulus) through the neural cluster level (e.g. modeling the release and effects of dopamine in the basal ganglia) to the complete organism (e.g. behavioural modeling of the organism's response to stimuli). Artificial intelligence cognitive modeling, and neural networks are information processing paradigms inspired by the way biological neural systems process data. Neural networks have been used in classification problems, speech recognition, stock market prediction and performance analysis, pattern matching, natural language procession, vision processing. Signal processing, image processing, decision making, optimization and control systems.

3.2.4.1. Characteristics of neural networks

- Learn by Example
- Mapping Capabilities
- Robust Systems
- Fault Tolerant
- Parallelism
- Flexibility
- Adaptive Learning
- Ability
- Self-Organization

3.2.4.2. Back Propagation

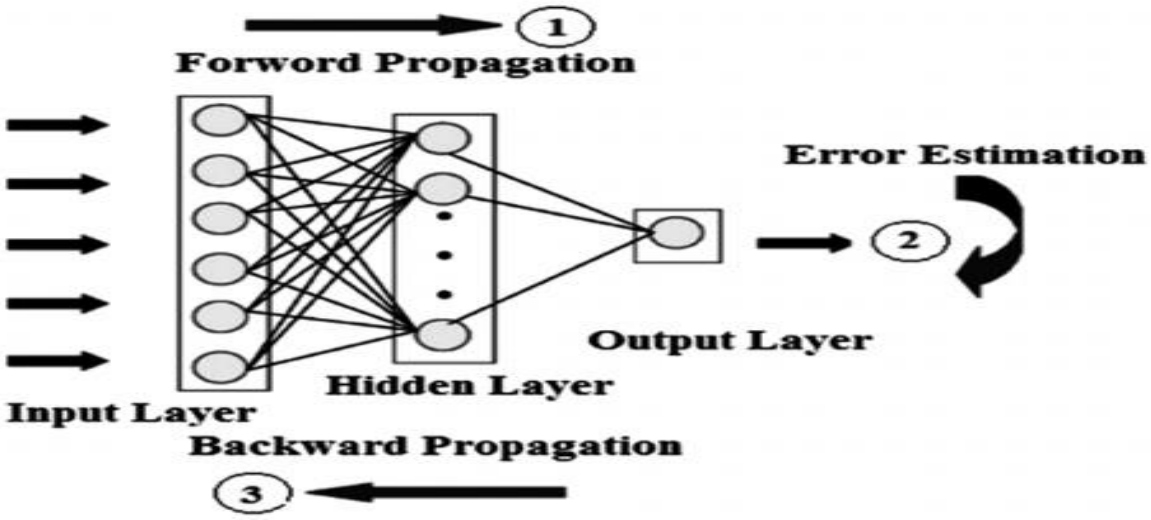

Fig. 3.2.4.2 Back Propagation

Back Propagation algorithm is used to train feed forward neural. It has two phases that is forward and backward pass phase. In forward pass phase, computes the function signal to get the output for input to hidden layer and hidden to output layer. In backward pass phase, computes the error signal to get the output for backward form.

Algorithm:

Step 1: Initialization

Step 2: Consider at random weight and input value.

Step 3: Do forward pass through the net to produce output.

- I/O applied
- Multiplied with weight
- Summarized (Produced net value)
- 'Squashed' by sigmoid activation function.
- Output passes to each neurons of the next layer.

Step 4: Computes the activation signal for input to hidden and hidden to output layer using step 2.

Step 5: Computes the error over the output neuron by comparing desired with actual output.

Step 6: Use the error to computes the change in hidden to output and input to hidden weights value

Step 7: Update the weight value.

Step 8: Repeat step 3-6 until network is not train.

3.2.5. Arduino

The Arduino UNO is a widely used open-source microcontroller board based on the ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board features 14 Digital pins and 6 Analog pins. It is programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable.[4] It can be powered by a USB cable or by an external 9 volt battery, though it accepts voltages between 7 and 20 volts. It is also similar to the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform. The ATmega328 on the Arduino Uno comes preprogrammed with a boot loader that allows to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol. The Uno also differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. The Arduino UNO is generally considered the most user-friendly and popular board, with boards being sold worldwide for less than 400Rs.

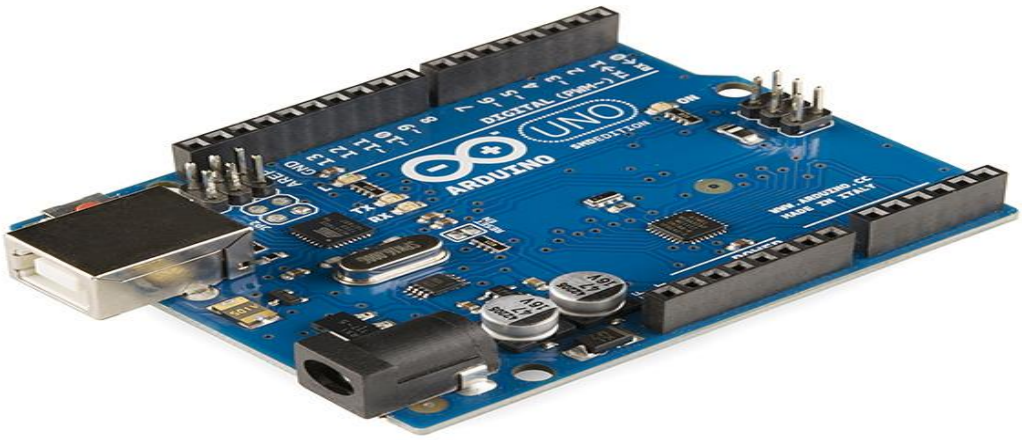

Fig. 3.2.5 Arduino Board

3.2.5.1. Features of the Arduino UNO:

- Microcontroller: ATmega328
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limits): 6-20V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6
- DC Current per I/O Pin: 40 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 0.5 KB used by bootloader
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Clock Speed: 16 MHz

3.2.5.2. Arduino Programming

At this point is fair to men on why speed parameters have to be modified:

When using the wired USB interface, the fastest data transmission speeds can be easily achieved, in this case the only limitation would be the processing speed on one end. Usually the speed used is 115000 bauds per second.

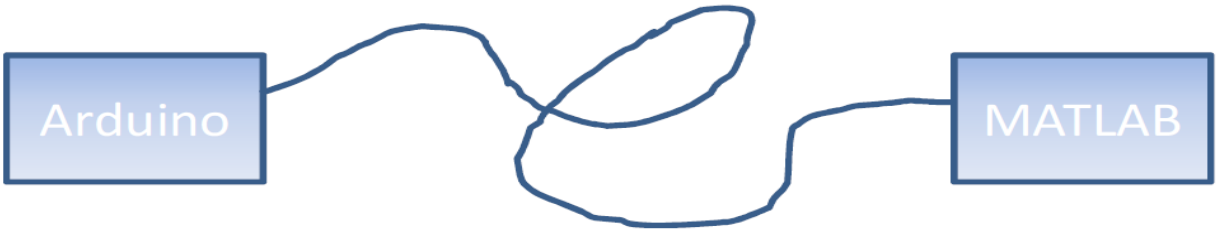

Fig. 3.2.5.2.1 USB Wire 115200 Bauds

When using the ZigBee adapter we face a speed limitation due to the ability of the Xbee to send and receive data, although 115200bauds speed can be achieved, it caused data loss and therefore erratic behavior of the rover.

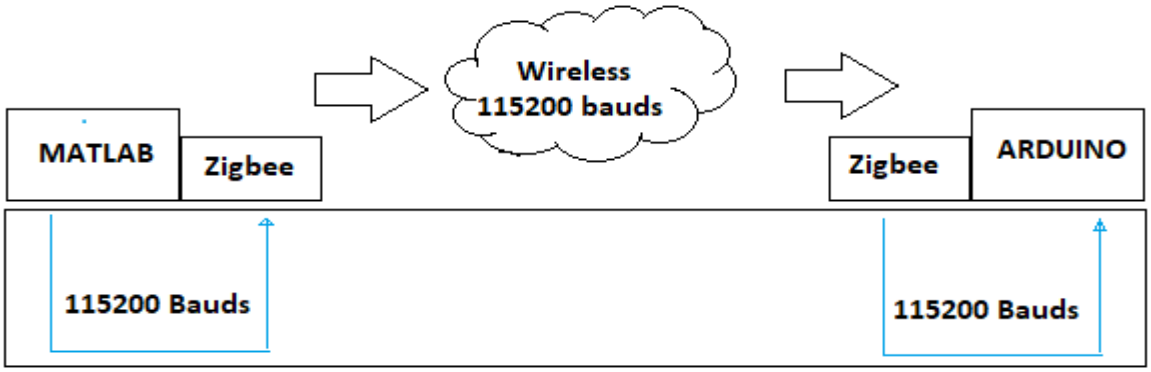

Fig.3.2.5.2.2 Wireless connection between MATLAB and Arduino

3.2.6. Motor Driver IC (L293D)

L293D is a dual H-bridge motor driver integrated circuit (IC). Motor drivers act as current amplifiers since they take a low-current control signal and provide a higher-current signal. This higher current signal is used to drive the motors.L293D contains two inbuilt H-bridge driver circuits. In its common mode of operation, two DC motors can be driven simultaneously, both in forward and reverse direction. The motor operations of two motors can be controlled by input logic at pins 2 & 7 and 10 & 15. Input logic 00 or 11 will stop the corresponding motor. Logic 01 and 10 will rotate it in clockwise and anticlockwise directions, respectively. Enable pins 1 and 9 (corresponding to the two motors) must be high for motors to start operating. When an enable input is high, the associated driver gets enabled. As a result, the outputs become active and work in phase with their inputs. Similarly, when the enable input is low, that driver is disabled, and their outputs are off and in the high-impedance state.

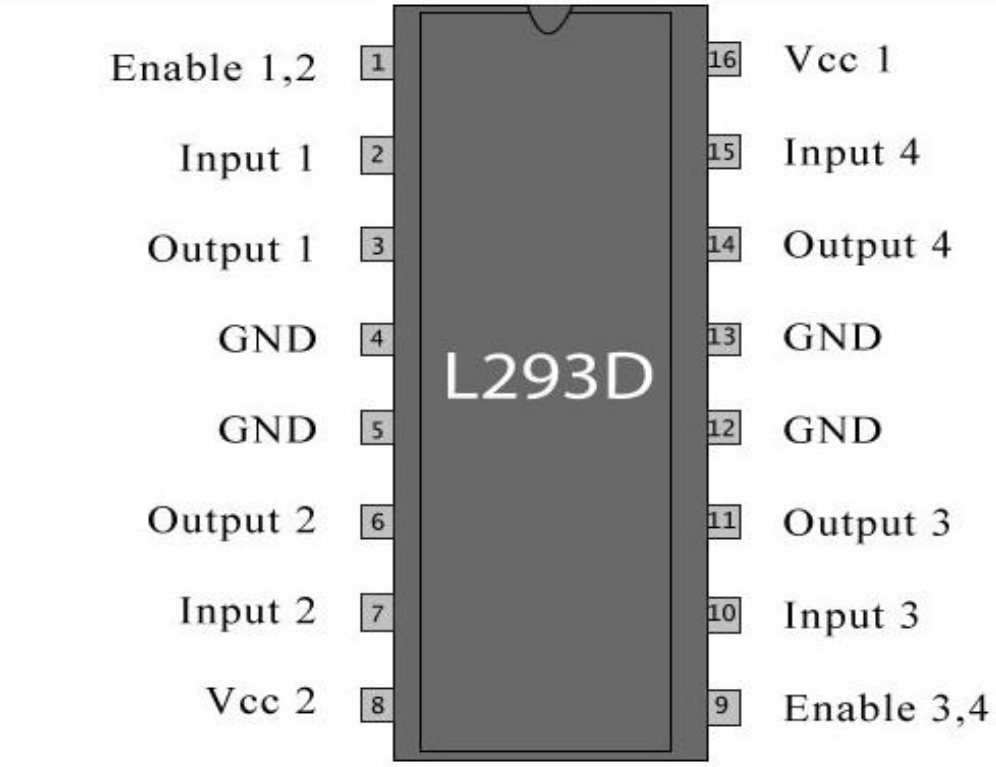

Fig. 3.2.6 Driver IC

3.2.7. DC Motor

A DC motor in simple words is a device that converts direct current (electrical energy) into mechanical energy. It's of vital importance for the industry today, and is equally important for engineers to look into the working principle of DC motor in details that has been discussed in this article. In order to understand the operating principle of dc motor we need to first look into its constructional feature.

The very basic construction of a dc motor contains a current carrying armature which is connected to the supply end through commutator segments and brushes and placed within the north south poles of a permanent or an electro-magnet as shown in the diagram below. Now to go into the details of the operating principle of dc motor it's important that we have a clear understanding of Fleming's left hand rule to determine the direction of force acting on the armature conductors of dc motor. Fleming's left hand rule says that if we extend the index finger, middle finger and thumb of our left hand in such a way that the current carrying conductor is placed in a magnetic field (represented by the index finger) is perpendicular to the direction

of current(represented by the middle finger), then the conductor experiences a force in the direction (represented by the thumb) mutually perpendicular to both the direction of field and the current in the conductor.

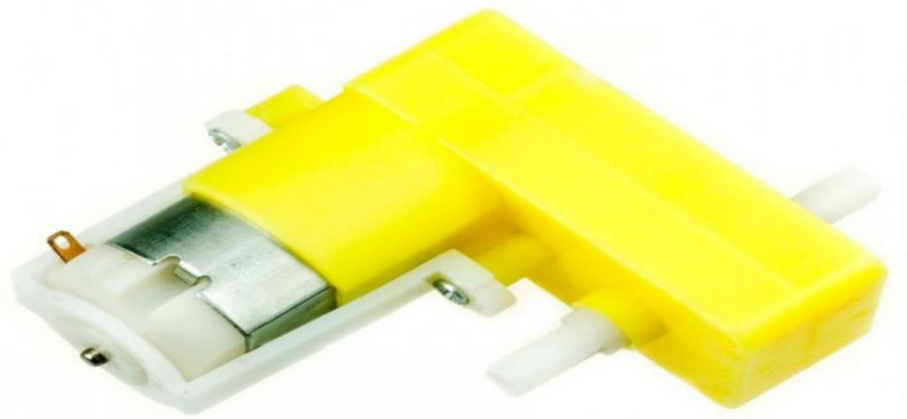

Fig. 3.2.7 DC Motor